Given a non-negative number, your task is to write this number in base 2 and calculate the sum of positions of 1s that are followed by 0s counting from right.

**Example**

For number = 83, the output should be  
majorIndex(number) = 7.

8310 = 1**01**0**01**12 which has 1's followed by 0's in positions 2 and 5 (reading from the right), so the answer is 2 + 5 = 7.

Positions of the digits to be found are colored red.

**Input/Output**

* **[time limit] 4000ms (py)**
* **[input] string number**

*Constraints:*  
0 ≤ int(number) < 253.

* **[output] integer**

<https://codefights.com/challenge/CN6zAWnaZYjcQTRyf?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

def **majorIndex**(number):

i = int(number)

b = bin(i)

sum =0

index =1

for i in range(len(b)-1,0,-1):

#print b[i]

if b[i] == *'1'* and b[i-1] == *'0'*:

sum += index

#print index

index+=1

return sum

print majorIndex(*"1000000"*)

---solucion por Alex Rafi--------

int r, i, j;

int majorIndex(string n)

{

//para convertir a binario

var m = Convert.ToString(Int64.Parse(n), 2);

j = m.Length;

for (i = j - 1; i > 0; i--)

if (m[i] == '1' & m[i - 1] == '0')

r += j - i;

return r;

}